REPRINT NO. 2

Matrix Programming of Electronic Analog Computers

Reprinted through the courtesy of
COMMUNICATIONS and ELECTRONICS
Published by the
AMERICAN INSTITUTE of ELECTRICAL ENGINEERS

★

Reprinted by
George A. Philbrick Researches, Inc.
Boston, Massachusetts

Reprinted through the courtesy of
COMMUNICATIONS and ELECTRONICS
Published by the
AMERICAN INSTITUTE of ELECTRICAL ENGINEERS

Reprinted by
GEORGE A. PHILBRICK RESEARCHES, INC.

Matrix Programming of Electronic Analog Computers

R. E. HORN
ASSOCIATE MEMBER AIEE

P. M. HONNELL
FELLOW AIEE

MATRIX PROGRAMMING of differential equations for electronic analog computers organizes the equations, thereby minimizing chances for errors; simplifies the scaling of coefficients and variables to suit the characteristics of the available machine; and in many instances permits the reduction of the number of "sign changers." The matrix method furthermore helps safeguard against the appearance of extraneous terms in the solutions of certain systems of differential equations. These very practical attributes of the matrix method of programming save computing time and equipment.

The key concept in the theory of matrix methods of programming electronic analog computers lies in the establishment of the 1-to-1 reciprocal correspondence between the matrically formulated differential equations and the synthesized electronic computer networks.[1-3] The generality of the matrix method leads not merely to one or two, but to a large group of electronic synthesis networks all yielding the solution to a given differential equation problem.

The object of this paper is to indicate by specific examples arising in practice the advantages of the matrix programming of electronic analog computers.

In the first example a wind-tunnel problem illustrates the organization, scaling of coefficients, and reduction of negative signs, thus minimizing the need for sign-changing amplifiers in a high-order system of differential equations.

This is followed by an elementary example exhibiting the care which must be taken in programming certain systems of differential equations to preclude the appearance of extraneous roots in the determinantal equation, and thereby extraneous solutions to the equations. This particularly insidious phenomenon may arise from devotion to the common "solve for the highest derivative" routine for synthesis of computer networks. These extraneous solutions are sometimes manifest as "drift" rather than in their true nature. Furthermore, the identical computer network sometimes yields "solutions" which are correct, and sometimes incorrect, depending upon the initial conditions. This is illuminated by a specific example.

Elementary Matrix Transformations

The objectives of some linear transformations of systems of linear algebraic equations and first-order differential equation systems of particular usefulness in analog computers are: 1. changing the magnitudes or signs of the coefficients of the terms in the equations, and 2. scaling in magnitude or changing the signs of the prescribed functions or dependent variables. Of course, such changes as are made must yield solutions unchanged or easily related to the original problem equations. The need for these transformations is often dictated by the purely electrical characteristics of the computer or recording equipment, such as the allowable output voltage range of the amplifiers. These elementary transformations are illustrated by the following example.

Example 1. The design of a wind-tunnel control system using analog computer techniques requires synthesis of the equations describing tunnel performance and the experimental determination of desirable controller characteristics. The particular wind-tunnel considered in this

Paper 58-66, recommended by the AIEE Basic Sciences Committee and approved by the AIEE Technical Operations Department for presentation at the AIEE Winter General Meeting, New York, N. Y., February 2-7, 1958. Manuscript submitted September 3, 1957; made available for printing November 7, 1957.

R. E. HORN is with Westinghouse Electric Corporation, Baltimore, Md., and P. M. HONNELL is with Washington University, St. Louis, Mo.

example may be adequately represented by the following differential equation system, in which P represents the pressures developed as a function of the prescribed variables, A and N_0. (The differential coefficient d/dt is represented by d.)

$$(1.90+d)P_2-0.423P_4-1.59P_{ac}=-0.303A_{12} \quad (1)$$

$$-0.433P_2+(4.51+d)P_4-1.87P_{md}-0.331P_m-4.16P_p=23.7A_{30}-1.29N_0 \quad (2)$$

$$-58.1P_2+(74.9+d)P_{ac}-30.8P_p=155N_0 \quad (3)$$

$$-1.01P_4+(0.124+d)P_{md}-7.52P_s=0.302N_0 \quad (4)$$

$$(0.309+d)P_{mi}-0.159P_m=37.8N_0 \quad (5)$$

$$-0.483P_{mi}+(0.617+d)P_m=-106.A_{30}-108.A_{33} \quad (6)$$

$$-0.00514P_2-0.091P_{ac}+(2.97+d)P_p-0.973P_s=0.173.A_{12}-2.18N_0 \quad (7)$$

$$-15.6P_{md}+(16.3+d)P_s=0 \quad (8)$$

$$151.P_p-54.3P_s+10^5P_n/P_p=0 \quad (9)$$

Valve openings which determine pressure are represented by A and are assumed to be controlled by functions of the differences between desired pressures P_{ms}, P_{ss}, or pressure ratio $(P_n/P_p)_s$, and respective measured values P_m, P_s, (P_n/P_p). Controller characteristics may be represented functionally by the following equations

$$A_{33}=f_1(P_{ms}-P_m) \quad (10)$$

$$A_{30}=f_2(P_{ss}-P_s) \quad (11)$$

$$A_{12}=f_3[(P_n/P_p)_s-(P_n/P_p)] \quad (12)$$

To illustrate application of the matrix technique to this problem, the synthesis procedure for this 8th-order system is described. Inclusion of controller characteristics involves an addition to the basic matrix equation and is not essential for the purpose of the present example.

An equivalent matrix representation of the system of equations 1 through 9 is indicated in equation 13 (see below).

The problem is to adjust the coefficients to reasonable values for use with a particular computer. Since the dependent variables are not known a priori, the term reasonable initially involves a "best guess" that tends to keep both the magnitude of the coefficients and dependent variables within acceptable limits. A convenient method is to apply transformations that attempt to establish coefficients approaching unity. For the majority of electronic analog computers, the step between the final mathematical form and the synthesis then involves the introduction of a factor of 10^{-6} in all terms, thus permitting immediate interpretation in terms of computer component values expressed in microfarads and micromhos, and computer variables expressed in volts and microamperes.

In equation 13 premultipliers affecting rows 3, 8, and 9, as follows:

row 3—divide by 100
row 8—divide by 10
row 9—divide by 100

result in reasonable values for all terms except that in row 9, column 9, which remains beyond normal limits. For cases of this type application of postmultipliers affecting columns in the coefficient matrix and also the dependent variables achieves the desired result. In particular, applying the column transformation

column 9—divide by 1,000

completes the coefficient adjustment necessary in equation 13.

Treatment of coefficients having an associated negative sign conventionally requires an additional amplifier to act as a sign changer. Reduction of the number of such sign changer is of advantage from both equipment and stability standpoints. Using the matrix formulation, transformations changing sign are easily applied and the effects of each transformation relative to the reduction of minus signs is immediately evident. Although unnecessary mathematically, electronic network stability usually dictates that transformation of rows and columns are made in pairs thus maintaining principal diagonal terms with positive sign. Desirable transformations for equation 12 change the signs of:

row 1, column 1
row 4, column 4
row 6, column 6
row 7, column 7
row 9, column 9

$$\begin{bmatrix} (1.90+d) & 0.423 & 1.59 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.433 & (4.51+d) & 0 & 1.87 & 0 & 0.0331 & 4.16 & 0 & 0 \\ 0.581 & 0 & (0.749+0.01d) & 0 & 0 & 0 & 0.308 & 0 & 0 \\ 0 & 1.01 & 0 & (0.124+d) & 0 & 0 & 0 & 7.52 & 0 \\ 0 & 0 & 0 & 0 & (0.309+d) & 0.159 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.483 & (0.617+d) & 0 & 0 & 0 \\ -0.00514 & 0 & 0.091 & 0 & 0 & 0 & (2.97+d) & 0.973 & 0 \\ 0 & 0 & 0 & 1.56 & 0 & 0 & 0 & (1.63+0.1d) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.51 & 0.543 & 1 \end{bmatrix} \begin{bmatrix} -P_2 \\ P_4 \\ P_{ac} \\ -P_{md} \\ P_{mi} \\ -P_m \\ -P_p \\ P_s \\ -10^3P_n/P_p \end{bmatrix} = \begin{bmatrix} 0.303A_{12} \\ 23.7A_{30}-1.29N_0 \\ 1.55N_0 \\ -0.302N_0 \\ 37.8N_0 \\ 106A_{30}+108A_{33} \\ -0.173A_{12}+2.18N_0 \\ 0 \\ 0 \end{bmatrix} \quad (14)$$

$$\begin{bmatrix} (1.90+d) & -0.423 & -1.59 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.433 & (4.51+d) & 0 & -1.87 & 0 & -0.0331 & -4.16 & 0 & 0 \\ -58.1 & 0 & (74.9+d) & 0 & 0 & 0 & -30.8 & 0 & 0 \\ 0 & -1.01 & 0 & (0.124+d) & 0 & 0 & 0 & -7.52 & 0 \\ 0 & 0 & 0 & 0 & (0.309+d) & -0.159 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.483 & (0.617+d) & 0 & 0 & 0 \\ -0.00514 & 0 & -0.091 & 0 & 0 & 0 & (2.97+d) & -0.973 & 0 \\ 0 & 0 & 0 & -15.6 & 0 & 0 & 0 & (16.3+d) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 151. & -54.3 & 10^5 \end{bmatrix} \begin{bmatrix} P_2 \\ P_4 \\ P_{ac} \\ P_{md} \\ P_{mi} \\ P_m \\ P_p \\ P_s \\ P_n/P_p \end{bmatrix} = \begin{bmatrix} -0.303A_{12} \\ 23.7A_{30}-1.29N_0 \\ 155.N_0 \\ 0.302N_0 \\ 37.8N_0 \\ -106.A_{30}-108.A_{33} \\ 0.173A_{12}-2.18N_0 \\ 0 \\ 0 \end{bmatrix} \quad (13)$$

The number of sign changers has now been reduced from 17 to one.

Results of the application of both scale-factor and sign-change transformations are indicated in equation 14 (see previous page), which is equivalent to equation 13 but more applicable to analog computer synthesis.

If equation 13 is represented as

$$[f_0][x_0]=[y_0] \tag{15}$$

and equation 14 as

$$[f_1][x_1]=[y_1] \tag{16}$$

equations 17 through 19 represent the elementary transformations.

$$[f_1]=\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.01 \end{bmatrix}[f_0]\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.001 \end{bmatrix} \tag{17}$$

$$[x_1]=\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1{,}000 \end{bmatrix}[x_0] \tag{18}$$

$$[y_1]=\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.01 \end{bmatrix}[y_0] \tag{19}$$

In view of their elementary character, the matrix premultipliers and postmultipliers in equations 17 through 19 need not be written down explicitly, although the information they contain, especially in the last, must be recorded.

The electronic network equation corresponding to equation 14 is indicated in equation 20, in which the coefficients now represent micromhos and microfarads, and the resultant ideograph is shown in Fig. 1.

Extraneous Solutions

The elementary solve-for-the-highest-derivative technique of computer programming, although restricted in form, usually gives correct solutions, particularly for single nth-order differential equations. In systems of differential equations, however, great care must be exercised to avoid extraneous solutions.

It is known that differentiation of equations in a system may introduce additional roots in the determinantal equation, and thus alien solutions. This constraint is easily overlooked in attempting to program a computer according to the currently accepted methods which require that each equation be written in such a manner that a derivative of a dependent variable appears to be a known function. Then, assuming the dependent variables are known, successive integrations are performed, functions combined, and eventually "fed back," completing the synthesis network. This common technique is not applicable to many systems of equations and the differentiation required in attempting to apply the method may easily yield erroneous results. Example 2 is an elementary illustration of this type of programming error.

Example 2. The following system of differential equations

$$\begin{aligned} &d^2x+d^2y-d^2z+dy+x+y=0\\ &dy+z=0\\ &2dx+dz+z=0 \end{aligned} \tag{21}$$

has the determinantal equation

$$\lambda(\lambda+1)(\lambda^2+1)=0 \tag{22}$$

and the general solution

$$\begin{aligned} &\dot{x}=\frac{(c_3-c_2)}{2}\cos t-\frac{(c_2+c_3)}{2}\sin t-c_4\\ &y=c_1e^{-t}-c_2\sin t+c_3\cos t+c_4\\ &z=c_1e^{-t}+c_2\cos t+c_3\sin t, \end{aligned} \tag{23}$$

in which the four constants specified in equation 23 are determined by the conditions

$$\begin{aligned} &x(0)=\frac{(c_3-c_2)}{2}-c_4\\ &y(0)=c_1+c_3+c_4\\ &z(0)=c_1+c_2\\ &dx(0)=-\frac{(c_2+c_3)}{2} \end{aligned} \tag{24}$$

It may appear desirable to differentiate the second and third equations of equation 21 yielding a set of equations which may then be programmed without difficulty using the conventional procedure.

The new system reads

$$\begin{aligned} &d^2x+d^2y-d^2z+dy+x+y=0\\ &d^2y+dz=0\\ &2d^2x+d^2z+dz=0 \end{aligned} \tag{25}$$

and has the determinantal equation

$$\lambda^3(\lambda+1)(\lambda^2+1)=0 \tag{26}$$

and the general solution

$$\begin{aligned} &x=\frac{(c_3-c_2)}{2}\cos t-\frac{(c_2+c_3)}{2}\sin t-(c_4+c_6)-c_6t\\ &y=c_1e^{-t}-c_2\sin t+c_3\cos t+c_4+c_6t\\ &z=c_1e^{-t}+c_2\cos t+c_3\sin t+c_5 \end{aligned} \tag{27}$$

$$\begin{bmatrix} (1.90+d) & 0.423 & 1.59 & 0 & 0 & 0 & 0 & 0 & 0\\ 0.433 & (4.51+d) & 0 & 1.87 & 0 & 0.033 & 4.16 & 0 & 0\\ 0.581 & 0 & (0.749+0.01d) & 0 & 0 & 0 & 0.308 & 0 & 0\\ 0 & 1.01 & 0 & (0.124+d) & 0 & 0 & 0 & 7.52 & 0\\ 0 & 0 & 0 & 0 & (0.309+d) & 0.159 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0.483 & (0.617+d) & 0 & 0 & 0\\ -0.005 & 0 & 0.091 & 0 & 0 & 0 & (2.97+d) & 0.973 & 0\\ 0 & 0 & 0 & 1.56 & 0 & 0 & 0 & (1.63+0.1d) & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1.51 & 0.543 & 1 \end{bmatrix}\begin{bmatrix} -w_1\\ -w_2\\ -w_3\\ -w_4\\ -w_5\\ -w_6\\ -w_7\\ -w_8\\ -w_9 \end{bmatrix}=\begin{bmatrix} j_1'\\ j_2'\\ j_3'\\ j_4'\\ j_5'\\ j_6'\\ j_7'\\ j_8'\\ j_9' \end{bmatrix} \tag{20}$$

Fig. 1. Ideograph of equation 20

The six constants in equation 27 are related to the initial conditions by the relations

$$
\begin{aligned}
x(0) &= \frac{(c_3-c_2)}{2} - c_4 - c_6 \\
y(0) &= c_1+c_3+c_4 \\
z(0) &= c_1+c_2+c_5 \qquad (28)\\
dx(0) &= -\frac{(c_2+c_3)}{2} - c_6 \\
dy(0) &= -c_1-c_2+c_6 \\
dz(0) &= -c_1+c_3
\end{aligned}
$$

Comparison of the solution in equation 23 with that in equation 27 clearly indicates the appearance of extraneous constant and linear terms in the latter.

In the general situation, the solutions to the differential equation system are not known explicitly (else why use the computer?). This leads to the following interesting situation. Assume that the conventional technique has been followed and the computer set up according to the differentiated system equation 25. Now suppose the transient response to the initial conditions

$$
\begin{aligned}
&x(0) = 20 \\
&y(0) = z(0) = dx(0) = 0 \qquad (29)
\end{aligned}
$$

is desired. Then the solutions given by equations 23 and 27 both read

$$
\begin{aligned}
x &= \frac{1}{3}(20 \cos t + 40) \\
y &= \frac{1}{3}(20e^{-t} + 20 \cos t + 20 \sin t - 40) \qquad (30)\\
z &= \frac{1}{3}(20e^{-t} - 20 \cos t + 20 \sin t),
\end{aligned}
$$

i.e., either system yields the identical solution.

Next, consider the solution for the initial conditions

$$
\begin{aligned}
&x(0) = y(0) = z(0) = 0 \\
&dx(0) = 20 \qquad (31)
\end{aligned}
$$

The solution in equation 23 to the original problem of equation 21 now yields

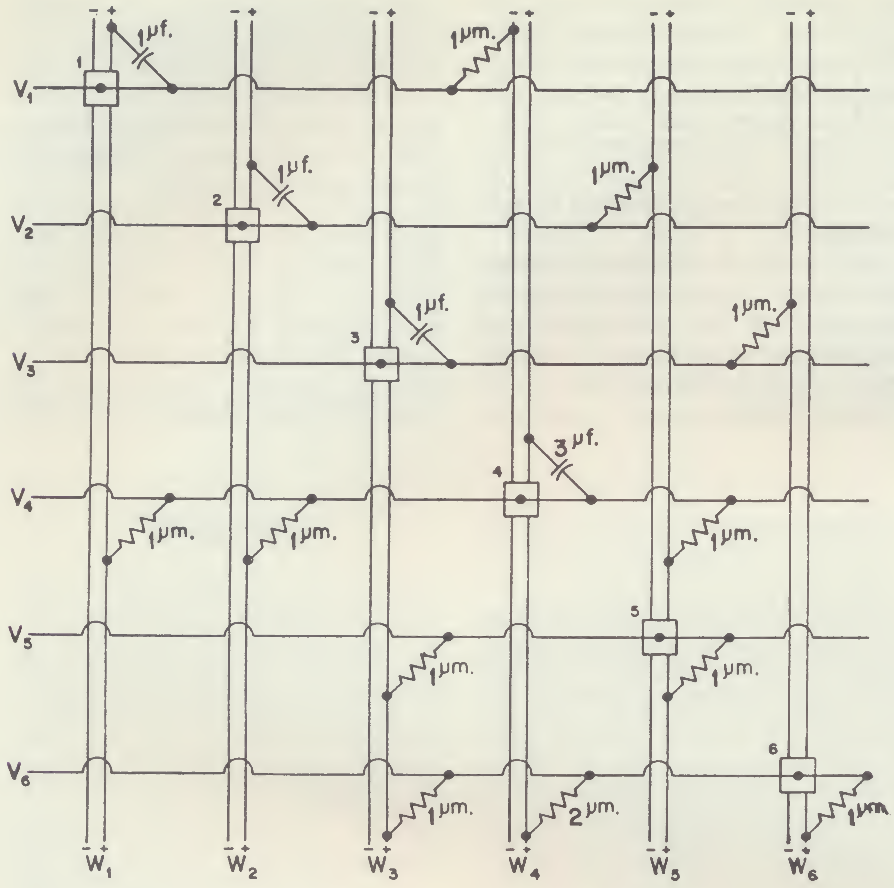


Fig. 2. Ideograph of equation 34

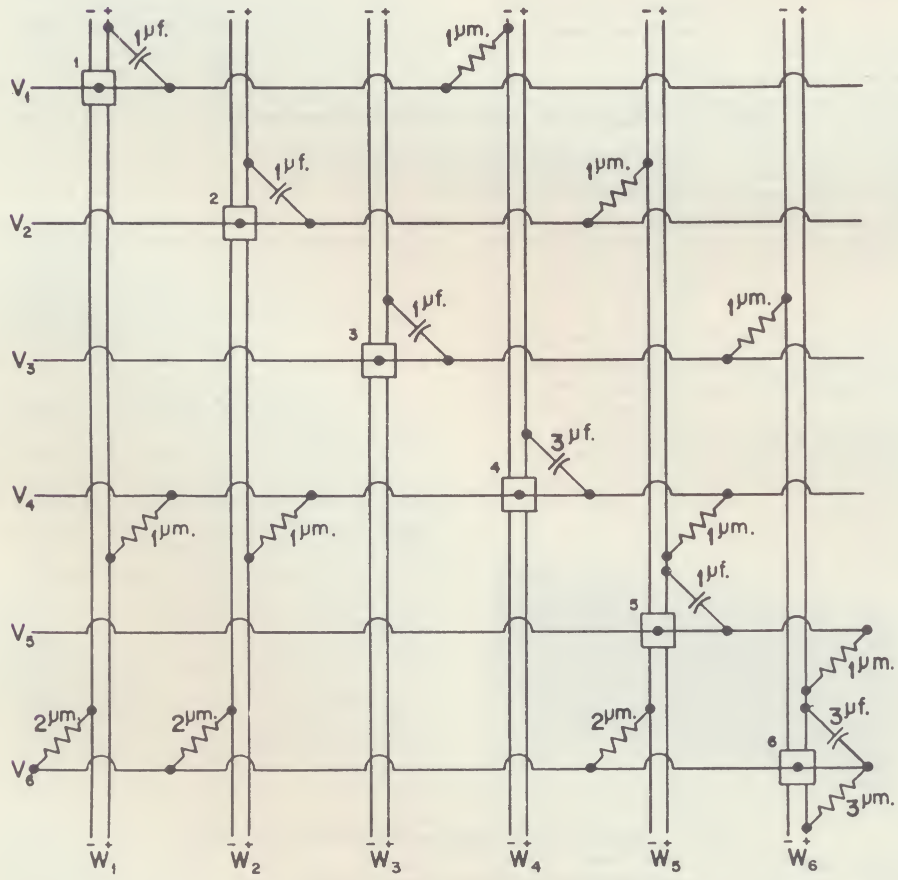


Fig. 3. Ideograph of equation 38

$$
\begin{aligned}
x &= 20 \sin t \\
y &= 20e^{-t} + 20 \sin t - 20 \cos t \qquad (32) \\
z &= 20e^{-t} - 20 \sin t - 20 \cos t
\end{aligned}
$$

whereas equation 27 yields the distinctly different solution

$$
\begin{aligned}
x &= \frac{1}{3}(20 \sin t + 40\,t) \\
y &= \frac{1}{3}(-20e^{-t} + 20 \sin t - 20 \cos t + 40 - 40t) \qquad (33) \\
z &= \frac{1}{3}(-20e^{-t} - 20 \sin t - 20 \cos t + 40)
\end{aligned}
$$

Recalling that the computer has been set up according to the common technique, i.e., according to equation 25, it is clear that the computing machine (which merely exhibits graphical plots) will in the first instance, equations 29 and 30, yield correct solutions to the ostensible problem in equation 21. In the second instance, equation 31, the computer exhibits solutions which do not satisfy the original problem in equation 21.

Of course, many other combinations of initial conditions are possible; but the machine never divulges which are indeed the correct solutions to the original problem, and which are not. This must be ascertained by other means. In this instance, the unwary may easily be misled, for as shown, sometimes the correct solutions are obtained, and sometimes they are not.

The matrix formulation of the problem, however, places in a clearer light the well-known mathematical requirements on transformations such that any synthesized computer network, if stable, yields the solutions to the original problem. This is indicated in the next section, together with oscillograms of computer solutions to the differential equation problem of equations 21 and 25 programmed by the matrix method.

Matrix Programming

The matrix programming of the differential system of equation 21 has been indicated and explained in reference 1, and one version leads to the equation

$$
\begin{bmatrix} d & 0 & 0 & -1 & 0 & 0 \\ 0 & d & 0 & 0 & -1 & 0 \\ 0 & 0 & d & 0 & 0 & -1 \\ 1 & 1 & 0 & 3d & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 2 & 0 & 1 \end{bmatrix}
\begin{bmatrix} -w_1 \\ -w_2 \\ -w_3 \\ -w_4 \\ -w_5 \\ -w_6 \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad (34)
$$

which is illustrated by the ideograph of Fig. 2. It therefore suffices to synthesize the differentiated system of equation 25.

According to the method outlined in the

the first reference, equation 25 may be written as

$$\left\{\begin{bmatrix}1&1&0\\0&0&0\\0&0&0\end{bmatrix}+\begin{bmatrix}0&1&0\\0&0&1\\0&0&1\end{bmatrix}D+\begin{bmatrix}1&1&-1\\0&1&0\\2&0&1\end{bmatrix}D^2\right\}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix} \tag{35}$$

in which x_1, x_2, and x_3 represent x, y, and z respectively.

One first-order system equivalent to equation 35 reads

$$\begin{bmatrix}d&0&0&-1&0&0\\0&d&0&0&-1&0\\0&0&d&0&0&-1\\1&1&0&d&(1+d)&-d\\0&0&0&0&d&1\\0&0&0&2d&0&(1+d)\end{bmatrix}\begin{bmatrix}-w_1\\-w_2\\-w_3\\-w_4\\-w_5\\-w_6\end{bmatrix}=\begin{bmatrix}0\\0\\0\\0\\0\\0\end{bmatrix} \tag{36}$$

Since this is likely to be unstable on existing analog computers, see reference 2, equation 36 may be premultiplied by

$$\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&-1&1\\0&0&0&0&1&0\\0&0&0&-2&2&1\end{bmatrix} \tag{37}$$

resulting in the matrix equation 38 which may be synthesized without difficulty and is stable.

$$\begin{bmatrix}d&0&0&-1&0&0\\0&d&0&0&-1&0\\0&0&d&0&0&-1\\1&1&0&3d&1&0\\0&0&0&0&d&1\\-2&-2&0&0&-2&3(1+d)\end{bmatrix}\begin{bmatrix}-w_1\\-w_2\\-w_3\\-w_4\\-w_5\\-w_6\end{bmatrix}=\begin{bmatrix}0\\0\\0\\0\\0\\0\end{bmatrix} \tag{38}$$

Fig. 3 is an ideograph of the electronic computer synthesis network corresponding to equation 38, and may be compared with Fig. 2.

Comparison of solutions exhibited by an analog computer for the two sets of differential equations corroborate the assertions of the previous section.

Figs. 4(A) and (B) respectively show the computer solutions to equations 21 and 25 programmed according to the matrix equations 34 and 38 for initial conditions of equation 29. As stated, the machine solutions coincide in this case.

For the initial conditions of equation 31, the oscillograms Figs. 5(A) and (B) correspond respectively to equations 32 and 33. As predicted, there is now no correlation between the respective solutions exhibited by the computer.

Résumé

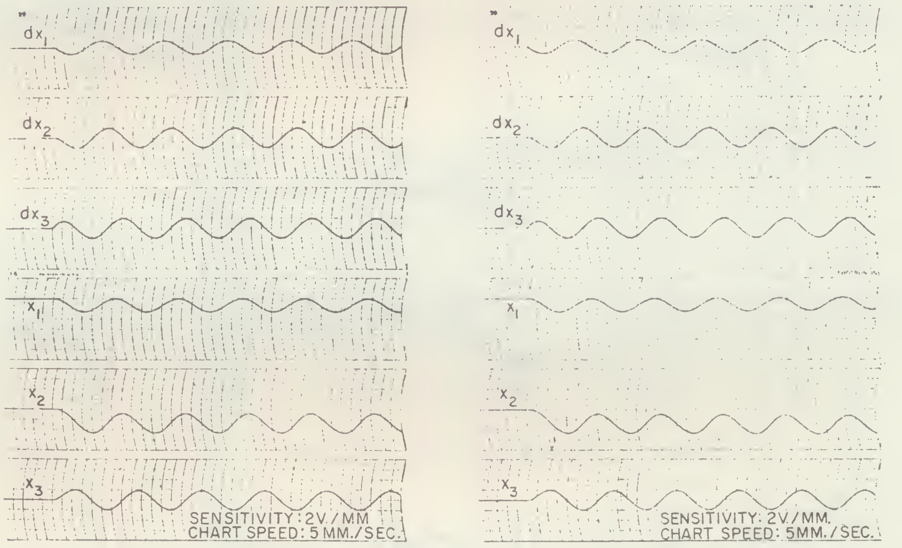
The practical merits of the matrix programming of analog computers have been briefly demonstrated in this paper. The most elementary of matrix transformations are an invaluable time- and equipment-saving process, as the first example indicates.

With regard to the second example and the introduction of extraneous roots in a system of differential equations, so strikingly demonstrated by the oscillograms, again matrix methods are extremely valuable.

It is well known that to obtain equivalent differential equation systems, the determinant of the transformation matrix must be a constant. In particular, the system of equation 38 may be obtained directly from equation 34, if the latter is premultiplied by the matrix

$$\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&-1&0&d&0\\0&0&-3&-2&0&3d\end{bmatrix} \tag{39}$$

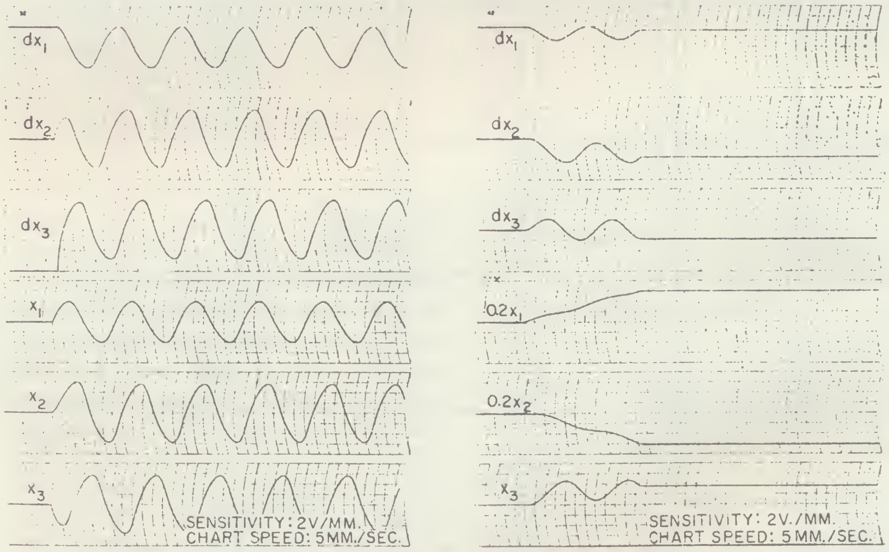
The determinant of equation 39 is $3d^2$.



(A) (B)

Fig. 4. Oscillograms for initial conditions of equation 31

A—Programmed according to equation 21
B—Programmed according to equation 25



(A) (B)

Fig. 5. Oscillograms for initial conditions of equation 31

A—Programmed according to equation 21
B—Programmed according to equation 25

Clearly this is not a constant and accounts for the extraneous λ^2 term in the determinantal equation 26, and additional terms in the solution equation 33.

In many practical problems which contain equations that have been differentiated for programming purposes, the manner of excitation of the network and initial conditions are such that the terms corresponding to the extraneous roots should have zero coefficients. However, whether due to initial or residual charges on the capacitors, grid current, internal parasitic coupling, or inexact matching of components, a small constant or linear term may be excited in addition to the correct solution response. The presence of such linear terms in the exhibited solution may then be interpreted as amplifier "drift," rather than actually as an incorrect synthesis network. This is the situation which may easily arise as indicated by the second example.

Finally, it should be clear that the application of matrix methods to analog computer programming are of far wider scope and generality than can be indicated in this brief introduction. But the practicality of the matrix method is clearly established. If not employed, the most economic operational use of the machine is not realized. This is the antithesis of the *raison d'être* for the computing machine in the first place!

References

1. Matrices in Analogue Mathematical Machines, Pierre M. Honnell, Robert E. Horn. *Journal*, Franklin Institute, Philadelphia, Pa., vol. 260, no. 3, Sept. 1955, pp. 193–207.

2. Matrices in Electronic Differential Analyzers, Pierre M. Honnell, Robert E. Horn. *Proceedings*, International Analogy Computation Meeting, Brussels, Belgium, 1956, pp. 217–21.

3. Analogue Computer Synthesis and Error Matrices, P. M. Honnell, R. E. Horn. *AIEE Transactions*, vol. 75, pt. I, Mar. 1956, pp. 26–32.

Discussion

Louis A. Pipes (University of California, Los Angeles, Calif.): Professor Honnell is one of the most outstanding leaders in stressing the importance of matrices and matrix calculus in the solution of problems in engineering and physics. Through his efforts and those of others it is now well-known that the theories of elastic structures, electric circuits, wave propagation, and mechanical vibration may be formulated concisely by the use of matrices, and that practical results may be obtained by means of the theorems of matrix algebra.

The development of modern instruments of computation has created new matrix problems. The utility of matrices and matrix algebra in the formulation and solution of many physical problems by the use of digital computing machines is now quite well known and is standard practice. Professor Honnell and Dr. Horn are to be congratulated for demonstrating the power and elegance of matrix methods in systematizing the programming of physical problems in analog computing machines.

It is to be hoped that when the simplicity and conciseness of the method presented by the authors are fully recognized, it will become a standard procedure in the programming of physical problems to be solved by analog computers.

Vincent C. Rideout (University of Wisconsin, Madison, Wis.): As more complex sets of equations are attempted on analog computers, systematic methods, both for scaling and for manipulation prior to setting up block diagrams, become more necessary. The matrix methods introduced in this paper provide a valuable guide to those confronted by such problems. Although presented for equations representing linear systems, they are also of help in nonlinear system study, if linearized forms of the system equations are first examined.

The introduction of extraneous roots by differentiation is clearly pointed out in the second example of the paper. However, it would be interesting to know if the equations of this example could come directly from a physical system. Also, this set may be shown to require only four arbitrary constants by the fact that it can be reduced to a third-order equation in y with the introduction of only one arbitrary constant in the process of this reduction.

Henry M. Paynter (American Center for Analog Computing, Boston, Mass.) and **Daniel H. Sheingold** (George A. Philbrick Researches, Inc., Boston, Mass.): The authors' use of the matrix approach is to be commended for the ease of scaling, the economy of amplifiers, and the inherent solution stability, all of which are direct consequences of a highly organized and well-thought-out formulation technique. The comprehensive treatment of the subject of redundant integrations and ambiguous solutions is very timely and should be of paramount importance to all those seriously interested in the machine solutions of systems of differential equations.

It is also particularly interesting to note the compact matrix form of the amplifier diagram; one novel feature of the symbolism is the use of conductance measures rather than resistances which results in a direct relationship between coefficients in the equations and physical parameters in the machine. However, we might suggest that a logical advantage could result from placing the feedback conductor in such ideographs in a position analogous to the capacitor whenever an amplifier is used merely as a summer or inverter. It is recognized, nevertheless, that there are perhaps comparable advantages to having all capacitors northwest/southeast and all conductors northeast/southwest.

The principal purpose of this discussion is to show that certain commercially available computers permit direct translation from the original matrix form (and, of course, from the original equations) to machine interconnections and parametric settings. Such machines are composed of modular elements which are called, for convenience, Universal Linear Operators. Fig. 6 shows the front panel of one modular element, and Fig. 7 shows a bank of 12 such elements. The basic equation, which completely describes the performance of each element is

$$e = e_0 + 10^m p^{-n} \sum_{i=1}^{4} a_i e_i \tag{40}$$

where

$m = 0, 1, 2, 3$

$n = 0, 1$

$$1/p = \int_0^t (\ )\,dt \tag{41}$$

The signal e_0 is adjustable to any positive or negative constant voltage within the output range of the element in steps of 0.1 volt;

Fig. 6. Front view of a Universal Linear Operator showing decade parameter settings and external connections

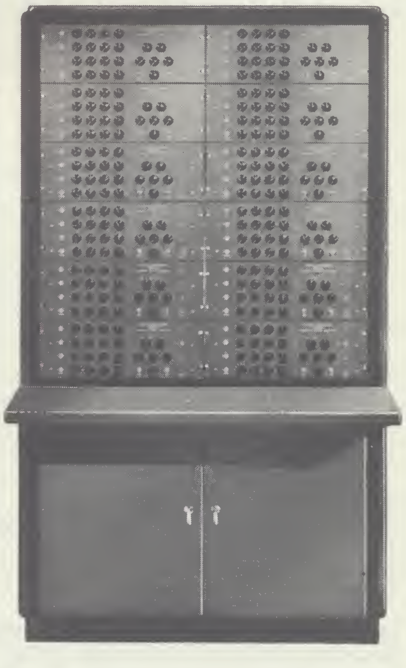


Fig. 7. A bank of 12 Universal Linear Operators

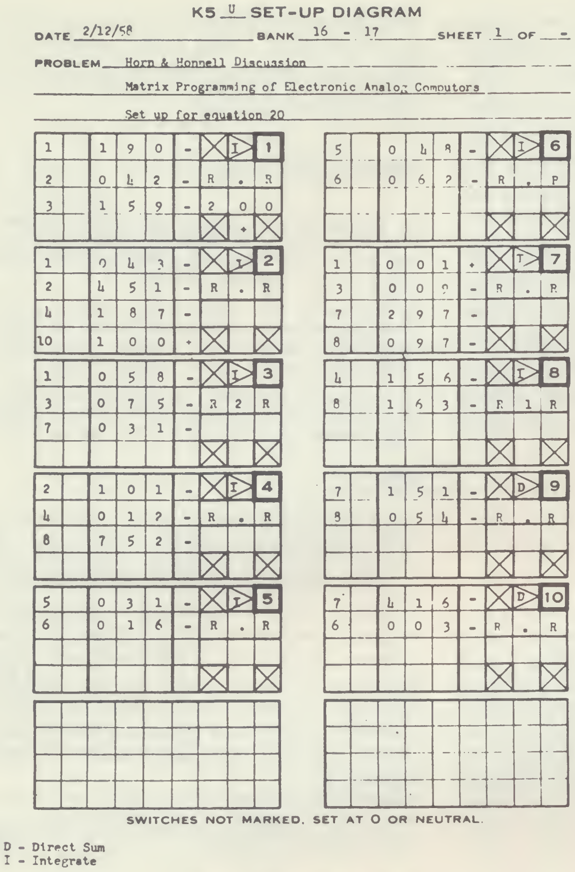


Fig. 8 (left). Tabular setup for equation 20

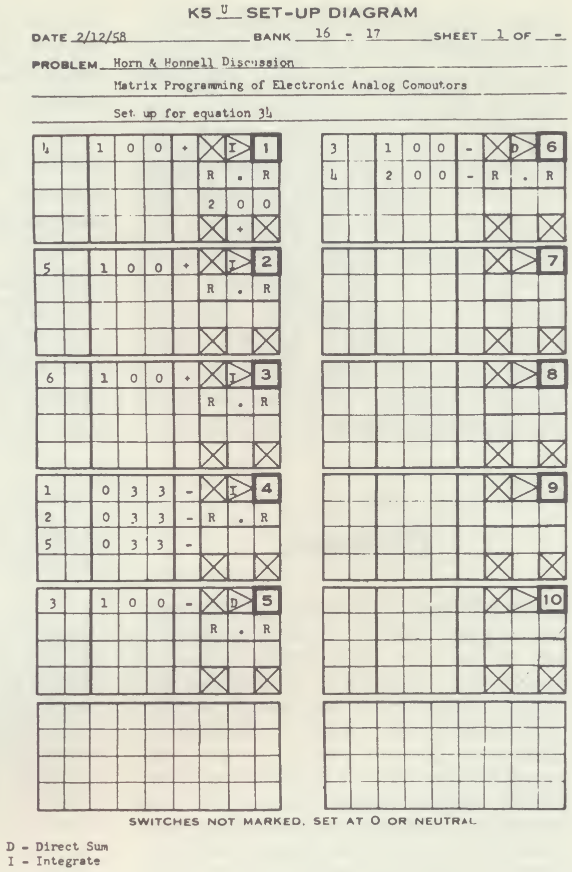


Fig. 9 (right). Tabular setup for equation 34

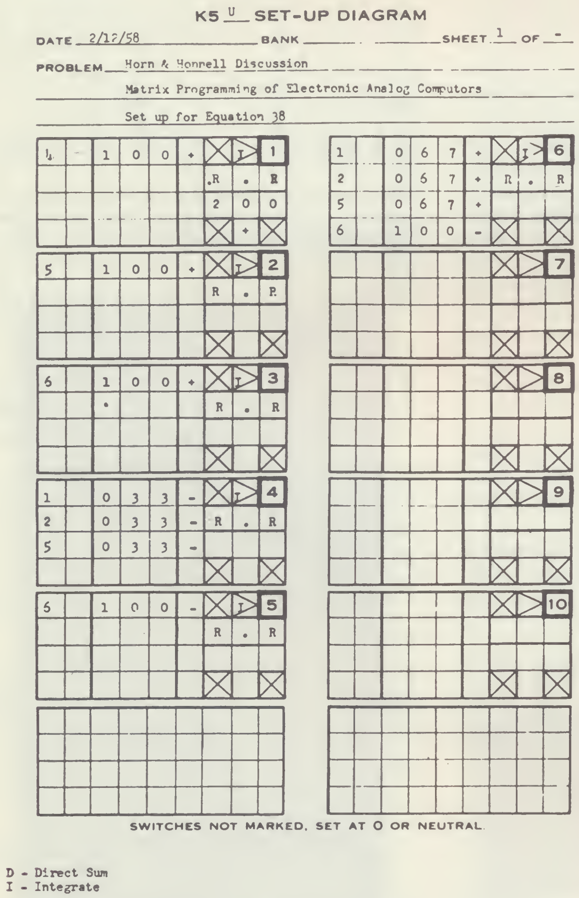


Fig. 10 (left). Tabular setup for equation 38

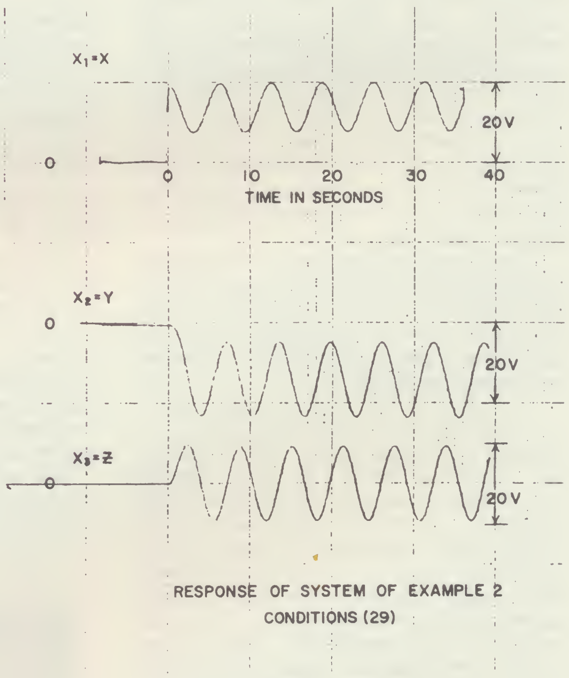


Fig. 11 (right). Solutions of equation 34 for x_1, x_2, and x_3

and a_i is adjustable to any positive or negative value from 0 to 11.10 in steps of 0.01 by means of 3-digit decade settings. It is seen immediately that this operator may function either as an integrator ($n=1$) or as a summer ($n=0$), and that the m parameter permits, in effect, a choice among four values of feedback resistance or capacitance. The voltage e_0 then functions either as a directly additive constant in the case of simple summation, or as an initial condition in the case of integration.

The only fundamental distinction affecting problem formulation between the operators described in the foregoing and operational summing points in the authors' matrix diagrams is in the location of the inverting amplifiers. The operational elements described here have outputs which are of one sense only, i.e., "direct," or positive.

The choice of signs is made via a switch at each input. This somewhat simplifies programming formulation because all variables may be defined as positive and be given sensed coefficients where operated upon. Although more than the minimum number of amplifiers might be used for a given problem configuration, the fact that they are inherently available in each operator means that there is, in reality, no "waste" of amplifiers.

We show in Figs. 8 through 10 tabular forms which completely define the computor problem setup including:

1. interconnections
2. parameters
3. initial conditions

and furthermore do so without reference to internal electronic functioning. Thus, individuals without prior knowledge of the nature of analog computing circuitry can successfully set up such programs directly from the matrix equations, and entirely nontechnical personnel can set up and run the machine directly from the tabular forms depicted.

Fig. 8 gives the tabular setup for equation 20 and Fig. 1 (for the homogeneous case). Note that the module numbers correspnd one to one with the problem variables (the columns in the matrix). Note also that when more than four inputs are required as in the second row of equation 20, additional modules can be employed; in this case only one extra module (number *10*) was required.

Figs. 9 and 10 correspond to equations 34 and 38 and Figs. 2 and 3 respectively. Note again that the parameters can be set directly from the matrix itself. Fig. 11 gives solutions of the second example under the initial conditions of equation 29, comparable to those shown in Fig. 4 for x_1, x_2, and x_3. If these three variables and their rates of change must be displayed then six operators must be used. However, if only the variables themselves are required, the minimum number of four operators may be used.

It may be of interest to note that the actual setup time requred by an individual reasonably experienced in the use of the equipment amounted to 4 minutes for equation 20, and 2 minutes apiece for the other two examples. The actual running time for each case being under 1 minute, the total time required for the solution of all three equations was less than 15 minutes.

V. H. Disney (Armour Research Foundation, Chicago, Ill.): The authors have developed a very clear and useful system for the orderly manipulation of the equations to be mechanized on an analog computer. Description of the equations in matrix form permits an easily checked layout of the programming connections which should do much to reduce the possibility of error. It should be noted, however, that only the scales of coefficients and dependent variables are changed by the procedure described. It is suggested that this system be extended to include change of time scale as well. If this were done the question arises as to what order of scale change in time, coefficient, and variable should be made for optimum programming. Guides might be developed for selecting reasonable values of the scale changes.

The authors have also described the difficulties that can arise when equations are differentiated. This is not a widely used technique. I should like to ask the authors if they would recommend avoiding this practice altogether, or whether there are means to test the solutions for validity.

R. E. Horn and **P. M. Honnell**: We are keenly appreciative of the favorable view of our work expressed by Dr. Pipes, for he is among the handful of mathematically inclined scientists in the entire world who have pioneered and steadfastly pursued researches for almost a quarter of a century in the application of matrix methods to engineering problems. This was long before the electrical engineering profession generally had any cognizance of the existence of these mathematic methods, much less of their application to electical engineering problems.

The writings of Dr. Pipes have, over the years been a source of instruction, guidance, and motivation to all of us who now pursue matric methods in our own specialties of the moment. In particular, we hope to be able to apply the results of the researches of Dr. Pipes, in the difficult area of variable and nonlinear systems, to the electronic computing techniques we describe; needless to say, these are matric methods! It must give Dr. Pipes great satisfactoin to see his prescience so abundantly substantiated.

The comments of Dr. Rideout concerning the application of these methods to systems of nonlinear equations are particularly pertinent. It is in these areas that a large percentage of problems of current interest lie—complex systems of equations in which the systematic matrix approach is particularly helpful. The authors plan to consider these aspects in more detail in their future publications.

With regard to Dr. Rideout's question concerning the origin of the equation in the second example, the basic form of the equation was obtained directly from a system of equations representing an aircraft hydraulic servo system. In the original simulation of this problem, the extraneous linear term was introduced inadvertently using conventional techniques and caused considerable difficulty in the manner that is described in the paper.

With reference to the discussion by Dr. Paynter and Mr. Sheingold the following applies. One of the objectives of the authors' paper was to indicate the application of the matrix approach to existing computers. By presenting such a complete discussion of the application to a particular commercial machine, the discussors have placed additional emphasis on this aspect of the use of matrix methods.

Because of their generality, the matrix programming methods should also aid considerably in obtaining the fullest utilization of other existing computers.

We are glad to have Mr. Disney bring out the question of change-of-time scaling. This is again most concisely stated in terms of premultipliers and postmultipliers. Consider, for example, the change in time scale represented by

$\tau/k = t$ and $f(\tau/k) = f(t)$,

where

t = original independent variable
τ = transformed independent variable
k = constant of proportionality

as applied to equation 22 of reference 3, namely

$$\begin{bmatrix} D & -I & 0 & \cdots & 0 & 0 \\ 0 & D & -I & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & D & -I \\ A_1 & A_2 & A_3 & \cdots & A_n & A_{n+1} \end{bmatrix} \begin{bmatrix} x \\ Dx \\ \cdot \\ D^{n-1}x \\ D^n x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ 0 \\ f(t) \end{bmatrix}$$

With appropriate premultipliers and postmultipliers (omitted herein), this yields the desired expression

$$\begin{bmatrix} D_\tau & -I & 0 & \cdots & 0 & 0 \\ 0 & D_\tau & -I & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & D_\tau & -I \\ A_1 & kA_2 & k^2A_3 & \cdots & k^{n-1}A_n & k^n A_{n+1} \end{bmatrix} \begin{bmatrix} x \\ D_\tau x \\ \cdot \\ D_\tau{}^{n-1}x \\ D_\tau{}^n x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ 0 \\ f(\tau/k) \end{bmatrix}$$

wherein

$D_\tau{}^n = I \times d_\tau{}^n$,

$d_\tau{}^n$ is written for $d^n/d\tau^n$, and, of course, $D = (1/k)D_\tau$.

With respect to Mr. Disney's last point, we defer to the mathematicians, who teach that equivalent differential systems can be obtained only if the determinant of the transformation matrices are constant. This permits the appearance of the derivative operator among the elements of transformation matrices, subjec to the aforementioned restriction. It does rule out the differentiation of equations per se.

1064A2 Printed in U.S.A.